

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Previously Presented) A database system comprising:
one or more storage devices containing a table having plural rows,
the plural rows including a first row containing a before image representing data of the first row of the table before a data modification operation on the first row and a second row containing an after image representing data of the first row of the table after processing by the data modification operation; and
a module, in response to a read request from a requester, to:
return the data in the second row to the requester under a normal condition, and
return the data in the first row to the requester under an abort condition in which the data modification operation is aborted, the abort condition causing a rollback operation involving the first and second rows to occur, the module to return data in the first row to the requester in response to the read request under the abort condition without having to wait for the rollback operation to complete.
2. (Original) The database system of claim 1, wherein the one or more storage devices further contain identifiers to identify a state of each row.
3. (Original) The database system of claim 2, wherein the identifiers are contained in the table.

4. (Previously Presented) The database system of claim 1, wherein each of the first and second rows are associated with a row identifier, the row identifiers of the first and second rows having a first value.

5. (Original) The database system of claim 4, wherein the table further contains state identifiers to identify a before image state of the first row and an after image state of the second row.

6. (Original) The database system of claim 1, wherein the data modification operation is performed in a transaction, the transaction having one or more requests, wherein the first row contains a transaction before image representing data before the beginning of the transaction, and wherein the plural rows further comprise a third row containing a request before image representing data before the beginning of a request in the transaction.

7. (Previously Presented) The database system of claim 5, wherein the module is adapted to transition the state of each row based on a data manipulation command.

8. - 9. (Cancelled)

10. (Previously Presented) The database system of claim 1, further comprising a rollback module adapted to perform the rollback operation and to mark the first row as containing a current image in response to the rollback operation.

11. (Previously Presented) The database system of claim 10, wherein the rollback module is adapted to further remove the second row in response to the rollback operation.

12. (Original) The database system of claim 1, wherein the table contains a first row identifier associated with the first and second rows, a first state identifier having a first value associated with the first row, and a second state identifier having a second value associated with the second row.

13. (Original) The database system of claim 12, wherein the table further contains a mutation identifier associated with the first row identifier to identify that the modification operation is occurring with respect to one or more rows associated with the first row identifier.

14. (Original) The database system of claim 13, wherein the mutation identifier changes value with each new modification operation.

15. (Previously Presented) A database system comprising:
one or more storage devices containing a table having plural rows,
the plural rows including a first row containing a before image representing data before a data modification operation and a second row containing an after image representing data processed by the data modification operation,
wherein the table contains a first row identifier associated with the first and second rows, a first state identifier having a first value associated with the first row, and a second state identifier having a second value associated with the second row,

wherein the table further contains a mutation identifier associated with the first row identifier to identify that the modification operation is occurring with respect to one or more rows associated with the first row identifier,

wherein the mutation identifier changes value with each new modification operation,

wherein the data modification operation is performed in a transaction, each transaction having one or more requests, the mutation identifier having a transaction identifier portion and a request identifier portion.

16. (Original) The database system of claim 15, wherein the transaction identifier portion has a value that increments with each new transaction.

17. (Previously Presented) The database system of claim 14, wherein the module is adapted to return a row based on the mutation identifier and state identifier information.

18. (Previously Presented) The database system of claim 14, wherein the one or more storage devices further contain an active mutation identifier list having plural mutation identifiers associated with respective plural active modification operations.

19. (Previously Presented) The database system of claim 18, wherein the one or more storage devices further contain an abort mutation identifier list having plural mutation identifiers associated with respective plural aborts of modification operations.

20. (Cancelled)

21. (Previously Presented) The method of claim 41, further comprising setting a first state identifier in the table to a first value to identify the first row as the before image row and setting a second state identifier in the table to a second value to identify the second row as the after image row.

22. - 23. (Cancelled)

24. (Currently Amended) ~~The method of claim 20, further comprising A~~
method of providing access in a database system, comprising:

storing data in rows of a table;

in response to a data modification operation of a first row, marking the first row as a before image row containing data of the first row before the start of the data modification operation on the first row, and creating a second row as an after image of the first row containing data processed by the data modification operation;

wherein the first and second rows are stored in the table;

in response to a read request from a requester, returning data from the first row to the requester under a first condition, and returning data from the second row to the requester under a second condition; and

rolling back to the first row if the data modification operation aborts, the first condition being an abort condition, wherein the first row is accessible for user request while the rollback is being performed.

25. (Original) The method of claim 24, further comprising deleting or marking as available for reuse the second row during a rollback process in response to the abort.

26. – 27. (Cancelled)

28. (Previously Presented) An article comprising at least one storage medium containing instructions that when executed cause a system to:

- store data in rows of a table;
- store a state identifier associated with each row, a first state identifier having a first value to indicate a first row as being a before image of a data modification operation on the first row of the table and a second state identifier having a second value to indicate a second row as being an after image of the data modification operation on the first row of the table,
- wherein the first and second rows are stored in the table; and
- in response to a read request from a requester,
 - returning data from the second row to the requester under a normal condition, and
 - returning data from the first row to the requester under an abort condition, the abort condition to cause a rollback operation to be performed, wherein returning data from the first row to the requester under the abort condition in response to the read request occurs without having to wait for the rollback operation to complete.

29. (Cancelled)

30. (Previously Presented) The database system of claim 1, wherein the table comprises a relational table for storing data of a database.

31. (Previously Presented) The database system of claim 1, further comprising: plural storage elements to store the table; and plural access module processors to enable parallel access of the plural storage elements.

32. (Cancelled)

33. (Previously Presented) The article of claim 28, wherein storing data in rows of the table comprises storing database data in rows of a relational table.

34. (Cancelled)

35. (Previously Presented) The database system of claim 1, wherein the data modification operation is responsive to a Structured Query Language (SQL) query, and the data of the first row is modified by the data modification operation responsive to the SQL query.

36. (Previously Presented) The database system of claim 35, wherein the read request is responsive to another SQL query.

37. (Previously Presented) The database system of claim 36, wherein the table further contains state identifiers associated with respective rows, the state identifier associated with the first row having a first value to indicate that the first row contains the before image, and the state identifier associated with the second row having a second value to indicate that the second row contains the after image.

38. (Previously Presented) The database system of claim 1, wherein the read request is invoked in response to a user request, and the first row is accessible by the user request even though the rollback operation is being performed, the module to return data in the first row to the requester under the abort condition in response to the read request while the rollback operation is occurring.

39. (Currently Amended) ~~The method of claim 20,~~ A method of providing access in a database system, comprising:
storing data in rows of a table;
in response to a data modification operation of a first row, marking the first row as a before image row containing data of the first row before the start of the data modification operation on the first row, and creating a second row as an after image of the first row containing data processed by the data modification operation;
wherein the first and second rows are stored in the table;
in response to a read request from a requester, returning data from the first row to the requester under a first condition, and returning data from the second row to the requester under a second condition;
wherein the first condition comprises an abort condition; and ,the method further comprising:
performing a rollback operation involving the first and second rows in response to the abort condition;
wherein returning data from the first row to the requester under the abort condition in response to the read request occurs without having to wait for the rollback operation to complete.

40. (Previously Presented) The method of claim 39, wherein returning data from the first row to the requester under the abort condition in response to the read request is performed while the rollback operation is occurring.

41. (Previously Presented) The method of claim 39, further comprising:
receiving a first Structured Query Language (SQL) query; and
invoking the data modification operation in response to the first SQL query.

42. (Previously Presented) The method of claim 41, further comprising:
receiving a second SQL query; and
invoking the read request by the requester. in response to the second SQL query.

43. (Previously Presented) The article of claim 28, wherein the instructions when executed cause the system to:
receive a first Structured Query Language (SQL) query; and
invoke the data modification operation in response to the first SQL query.

44. (Previously Presented) The article of claim 43, wherein the instructions when executed cause the system to:
receive a second SQL query; and
invoke the read request by the requester in response to the second SQL query.

45. (Previously Presented) The article of claim 28, wherein returning data from the first row to the requester under an abort condition in response to the read request is performed while the rollback operation is occurring.